

THE ICSTM+TUM+UP APPROACH TO THE 3RD CHiME CHALLENGE: SINGLE-CHANNEL LSTM SPEECH ENHANCEMENT WITH MULTI-CHANNEL CORRELATION SHAPING DEREVERBERATION AND LSTM LANGUAGE MODELS

Amr El-Desoky Mousa¹, Erik Marchi², Björn Schuller^{1,3}

¹Chair of Complex & Intelligent Systems, University of Passau, Passau, Germany

²Machine Intelligence & Signal Processing group, MMK, Technische Universität München, Munich, Germany

³Department of Computing, Imperial College London, London, UK

ABSTRACT

This paper presents our contribution to the 3rd CHiME Speech Separation and Recognition Challenge. Our system uses Bidirectional Long Short-Term Memory (BLSTM) Recurrent Neural Networks (RNNs) for Single-channel Speech Enhancement (SSE). Networks are trained to predict clean speech as well as noise features from noisy speech features. In addition, the system applies two methods of dereverberation on the 6-channel recordings of the challenge. The first is the Phase-Error based Filtering (PEF) that uses time-varying phase-error filters based on estimated time-difference of arrival of the speech source and the phases of the microphone signals. The second is the Correlation Shaping (CS) that applies a reduction of the long-term correlation energy in reverberant speech. The Linear Prediction (LP) residual is processed to suppress the long-term correlation. Furthermore, the system employs a LSTM Language Model (LM) to perform N-best rescoring of recognition hypotheses. Using the proposed methods, an improved Word Error Rate (WER) of 24.38% is achieved over the real eval test set. This is around 25% relative improvement over the challenge baseline.

Index Terms— Speech enhancement, dereverberation, correlation shaping, LSTM, language model

1. INTRODUCTION

Automatic Speech Recognition (ASR) in real-world noisy and reverberated environments is a challenging problem. The 3rd CHiME Speech Separation and Recognition Challenge (CHiME-3) addresses this problem by providing a testing platform for speech enhancement and recognition techniques [1]. The CHiME-3 scenario involves performing ASR for 6-channel microphone array data recorded via a multi-microphone tablet device being used in everyday, noisy environments. The data involves real acoustic mixing in four various noise settings, namely café, street junction, public transport and pedestrian area. The recordings are divided into training, dev and eval sets. Each set features different talkers and different instances of the same noise environment.

Methods for robust recognition of noisy speech can be categorized into two broad categories: the first category involves **front-end** enhancement of either the waveforms or the extracted features by removing noise and reverberation [2]. It is also possible to employ feature adaptations to transform the corrupted features, or to use noise-robust features directly. The other category involves improved **back-end** ASR systems. One method is to adapt the models to noisy features, e.g., using multi-condition training or methods such as vector Taylor series [3]. In particular, Deep Neural Network (DNN) models have shown robust performance in recent years for both acoustic models [4] and language models [5, 6, 7, 8].

Neural networks for blind non-linear source separation have been extensively studied; e.g. in [9, 10]. However, these works do not consider speech and noise model training. Training of ASR feature enhancement models has been considered in [11]. Therein, RNN Auto-Encoders (AE) are used to enhance cepstral-domain speech recognition features, but synthesis of time-domain signals is not considered. In [12, 13], a similar approach is considered with LSTM architectures and is found superior to standard RNNs. In [14], DNNs are used to map noisy to clean Mel-features, but the network outputs are synthesized directly into a time domain signal, instead of constructing a filter based on speech and noise magnitudes. In [15], a combination of unsupervised noise estimation and DNN based speech power spectrum estimation is used to construct a Wiener filter. However, this work does not consider learning noise models. In [16], DNNs are considered to predict the ideal ratio mask in an uncertainty decoding framework for ASR. However, the authors do not evaluate their models in terms of separation quality. In this paper, we apply an approach that we previously described in [17]. Thus, we use LSTM RNNs to model speech and noise features in a speech enhancement framework.

On the other side, reverberation severely degrades the performance of ASR. Suitable schemes for modeling reverberation are broadly applied such as the source-image method [18, 19]. Generally, a reverberant scenario consists of a source speech signal which propagates through an acoustic channel

and is then captured by a microphone. The microphone signal, however, contains a reverberated version of the source signal. Thus, dereverberation techniques are applied on the microphone signal to output an estimate of the source signal [3]. Many dereverberation algorithms have been developed over the last two decades [19]. Several strategies have been proposed, ranging from LP residual processing [20] to multiple microphone array based techniques [21, 22]. Further approaches have addressed blind system identification [23] by using subspace decomposition [24] and adaptive filters [25]. In this paper, we apply two multi-channel dereverberation techniques that we previously introduced in [3]. The first technique is called Phase-Error based Filtering (PEF). It relies on time-delay estimation with time-frequency masking [26, 27]. The second technique is called Correlation Shaping (CS) [28]. It is based on LP and reduces the length of the equalized speaker-to-receiver impulse response. Both techniques are applied on top of the enhanced recordings.

As a recognition back-end, our system relies on the baseline GMM/DNN acoustic models [1]. However, we employ a state-of-the-art LSTM LM to rescore the N-best recognition hypotheses obtained using the provided standard trigram LM.

2. EVALUATION DATABASE

The CHiME-3 scenario described in [1] involves ASR for a multi-microphone tablet device. 4 various environments are selected: caf (CAF), street junction (STR), public transport (BUS) and pedestrian area (PED). For each environment, two types of noisy speech data are provided, real and simulated. The real data consists of new 6-channel recordings of sentences from the WSJ0 corpus spoken in noisy environments. The simulated data is constructed by mixing clean utterances from that corpus into background recordings in the four noisy environments. For ASR evaluation, the data is divided into official training, dev and eval sets.

3. METHODOLOGY

3.1. Recurrent neural networks

3.1.1. Standard RNN architectures

The neural network architecture adopted for our Single-channel Speech Enhancement (SSE) model as well as our LM is based on LSTM RNNs [29]. A RNN can be described as an automaton-like structure mapping from a sequence of observations to a sequence of output features. These mappings are defined by activation weights and a non-linear activation function as in a standard Multi-Layer Perceptron (MLP). However, recurrent connections allow to access activations from past time. For an input sequence x_1^T , a RNN computes the hidden sequence h_1^T and the output sequence y_1^T by performing the following operations for $t = 1$ to T :

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where \mathcal{H} is the hidden layer activation function, W_{xh} is the weight matrix between input and hidden layer, W_{hh} is the recurrent weight matrix between hidden layer and itself, W_{hy} is the weight matrix between the hidden and output layer, b_h and b_y are the hidden and output layer bias vectors respectively.

In a standard RNN, \mathcal{H} is usually an element-wise application of sigmoid function. Such a network is usually trained using the Back-Propagation Through Time (BPTT) training [30], where a recurrent network with N timesteps is considered as an unfolded deep Feed-Forward Neural Network (FFNN) with N hidden layers and the error is propagated recursively from the hidden layer of the current timestep to the hidden layer of the previous timestep. However, the error gradients can quickly vanish as they get propagated in time or in rare cases grow exponentially [31]. This is known as the vanishing gradient problem. One solution to this problem is to consider only several steps of unfolding (truncated BPTT).

3.1.2. LSTM RNN architectures

In [32], an alternative RNN called Long Short-Term Memory (LSTM) RNN is introduced where the conventional neuron is replaced with a so-called *memory cell* that can be controlled by input, output and reset operations [33]. The purpose of this memory cell is to store information in such a way that the corresponding gradient is properly scaled and never gets lost. This has been shown to overcome the vanishing gradient problem of traditional RNNs [34]. Figure 1 illustrates the architecture of a single LSTM memory cell. In this case, \mathcal{H} can be described by the following composite function:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

where σ is the sigmoid function. i, f, o and c are respectively the input, forget, output gates and cell activation vectors. The weight matrices from cell to gate vectors are diagonal [29].

3.1.3. Deep architectures

A major factor in the recent success of neural network models is the use of deep architectures obtained by stacking multiple hidden layers on top of each other. In [35, 36], deep FFNNs have been successfully used for acoustic modeling in ASR tasks. Also, in [29, 37], both deep conventional and LSTM RNNs have been used for acoustic modeling in ASR.

Despite their context-sensitive nature, LSTM RNNs are well suited for online speech enhancement since they only require storing the current state of the automaton. In case that real-time capability is not needed, future context can also be exploited by adding a second set of layers which process the input feature sequences backwards, from $t = T$ to $t = 1$. This extension leads to Bidirectional LSTM (BLSTM) RNNs. In a deep BLSTM RNN, activations from both directions are

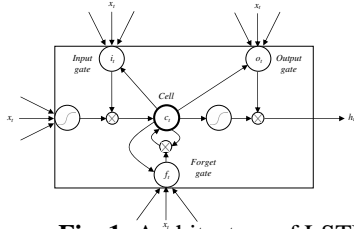


Fig. 1. Architecture of LSTM memory cell.

collected in a single activation vector before passing them on as inputs to the next layer [29].

3.2. LSTM RNN language model

In a LSTM LM [8], the timesteps correspond to the word positions in a training sentence. At every timestep, the network takes as input the word at the current position encoded using the 1-of- N binary encoding, where N is the vocabulary size. An optional projection layer with linear (or sigmoidal) activation function is used to project the binary word vectors into vectors of real values representing points in continuous space. The projected vectors, or directly the binary vectors, are passed to one or more recurrent hidden layers with self connections that implicitly take into account all the previous history words presented to the network. The output of the final hidden layer is passed to an output layer with a soft-max activation function to produce a correctly normalized probability distribution. The target output at each word position is the next word in the sequence. The training is based on the cross entropy criterion which is equivalent to directly maximizing the likelihood of the training data, consequently minimizing the LM perplexity. Thus, for a given history w_1^{n-1} the network can effectively predict the long-span conditional probability $p(w_n|w_1^{n-1})$ for any word w_n of the vocabulary.

3.3. Single-channel speech enhancement

Our SSE methodology is based on magnitude domain spectral subtraction. Let $\mathbf{X} \in \mathbb{R}^{F \times T}$ denote the magnitude spectrogram of a noisy speech signal with F discrete Fourier frequency bins and T observation frames. From \mathbf{X} , a clean

speech estimate \mathbf{Y} is computed through:

$$\mathbf{Y} = \mathbf{X} \otimes (1 - \hat{\mathbf{N}}/\hat{\mathbf{X}}) \quad (8)$$

where \otimes denotes element-wise multiplication and division is also element-wise. For traditional spectral subtraction, $\hat{\mathbf{X}} = \mathbf{X}$, so that the noise estimate is subtracted from the original noisy speech. Unsupervised estimation of $\hat{\mathbf{N}}$ is often done using minimum statistics [38].

Our data-based speech enhancement algorithm uses an additional clean speech estimate $\hat{\mathbf{S}}$ in the above filter, such that $\hat{\mathbf{X}} = \hat{\mathbf{S}} + \hat{\mathbf{N}}$. Thereby, the contribution of clean speech and noise to the observed signal can be predicted. Popular models for speech include non-negative (sparse) coding by non-negative matrix factorization or Hidden Markov Models (HMMs). In our system, we use LSTM RNN based modeling via supervised training of feature mappings similar to the denoising auto-encoder paradigm [15]. Due to their recent success in noise robust ASR [12, 39], RNNs appear to be very well suited to capture the dynamics of speech and noise as they directly model long-range context which cannot be approximated by ‘feature frame stacking’ in the general case.

In our system, networks are pre-trained to predict speech features from noisy speech features. Realistic instead of white Gaussian noise is used for training like in [14]. Similarly, networks are trained to predict noise from a convolutive mixture of speech and noise. During denoising, these estimates are used to construct a magnitude domain filter as in Equation 8. As features for the neural networks, we use log Mel-scale spectrograms $\mathbf{X}' \in \mathbb{R}^{B \times T}$ with $B = 40$ frequency bands equally spaced on the Mel-frequency scale. Thus, both amplitude and frequency are on a log-scale. These features have been proved highly successful for ASR with deep LSTM RNNs [29]. Given predicted log Mel-features of speech and noise, $\hat{\mathbf{S}}'$ and $\hat{\mathbf{N}}'$, the final filter equation is given by:

$$\mathbf{Y} = \mathbf{X} \otimes \left[1 - \frac{\mathbf{M}^{-1} \exp(\hat{\mathbf{N}})}{\mathbf{M}^{-1} (\exp(\hat{\mathbf{S}}') + \exp(\hat{\mathbf{N}}'))} \right] \quad (9)$$

where \mathbf{M}^{-1} denotes the ‘back-transformation’ from Mel to magnitude spectra and exponentiation is element-wise. Using Mel-spectra instead of magnitude or power spectra reduces the amount of speech features to be estimated. By reverting the Mel-scale transformation in the filter estimation, not in the estimated speech spectrogram, we avoid a loss of information due to the compression of the frequency axis. It is found that using the ‘ideal’ filter computed from ‘ground truth’ speech and noise Mel-spectra provided perfect reconstruction in many cases. This enhancement method is evaluated in our previous publication [17].

3.4. Dereverberation

3.4.1. Phase-error based filtering

Phase-Error based Filtering (PEF) involves time-varying, or time-frequency, phase-error filters based on estimated Time-Difference of Arrival (TDOA) of the speech source and the

phases of the microphone signals. The phase variance between two signals is defined as:

$$\psi_\beta = \sum_{k=1}^N \sum_{\omega=-\omega_s}^{\omega_s} \theta_{\beta,k}^2(\omega), \quad (10)$$

where

$$\theta_{\beta,k}(\omega) = \angle X_{1,k}(\omega) - \angle X_{2,k}(\omega) - \omega\beta \quad (11)$$

indicates the level of noise and reverberation present in the speech signal. $\angle X_{1,k}$ and $\angle X_{2,k}$ are the phase spectra of the input signals at frame k , and $\theta_{\beta,k}(\omega)$ is the minimized phase-error (PE) when β equals the TDOA, N indicates the number of segments in the speech signal, and ω_s is the highest frequency of interest. The PE measures the time misalignment at each frequency bin. The overall PE can be reduced to:

$$\theta_{\beta,k}(\omega) = \angle X_{1,k}(\omega) - \angle X_{2,k}(\omega) \quad (12)$$

with the assumption that the input signals are time-aligned. The PE is used as a reward-punish criteria to removing noise from multi-microphone speech signals. Time-frequency blocks with large PE are scaled down in amplitude, whereas, blocks with low PE are preserved. First, the PE is computed from the two phase spectra. Then, a masking function is applied as a weighting function for the amplitude spectrum of each channel. Spectra are later summed up similarly as delay-and-sum. The parametrized scaling function:

$$\eta(\omega) = \frac{1}{1 + \gamma \theta_{\beta,k}^2(\omega)} \quad (13)$$

is used in as a masking function to attenuate the time-frequency blocks, where γ is a fixed value. Higher values of γ reduce high PE blocks prominently with a consequent improved performance in low Signal-to-Noise Ratio (SNR) scenarios and worse performance in high SNR situations. PE based filtering is transferred to multi-microphone signals by applying the parametrized scaling function on all possible pairs of microphones. Each microphone pair i and j is processed by the masking function:

$$\eta_{ij}(\omega) = \frac{1}{1 + \gamma \theta_{ij}^2(\omega)} \quad (14)$$

extended from Equation 15. Then, a modified geometric mean of the time-varying functions [26] is used as follows:

$$\Phi_i(\omega) = \left[\prod_{j=1, \dots, M \wedge i \neq j} \eta_{ij}(\omega) \right]^{\frac{1}{m}} \quad (15)$$

where M is the number of microphones and m is a factor that affects the aggressiveness of the algorithm. For a standard geometric mean, $m = M$. Using this approach, the estimation of high PE values is relevant in the mask averaging process. In fact, provided that a pair of microphones results in a very high PE for a certain time-frequency block, the resulting scaling value will be close to zero. The zero value is then kept in the geometrical averaging with the masking values for other pairs of microphones. The enhanced spectrum $\hat{S}(\omega)$ is obtained by summing up the enhanced spectra processed by the multi-channel mask $\Phi_i(\omega)$, as follows:

$$\hat{S}(\omega) = \sum_{i=1}^M \Phi_i(\omega) X_i(\omega). \quad (16)$$

A detailed description and evaluation of the PEF dereverberation method is found in our previous publication [3].

3.4.2. Correlation shaping

Correlation Shaping (CS) reduces the long-term correlation in the LP residual of reverberant speech. This approach improves both the audible quality and ASR accuracy of reverberant speech [28]. CS modifies the correlation structure of the processed speech signal y . Assuming that an array of M microphones records a speech source, the signal observed by the m th microphone x_m is processed by an adaptive linear filter g_m in order to minimize the weighted Mean Square Error (MSE) between the actual output autocorrelation sequence \mathbf{R}_{yy} , and the desired output autocorrelation sequence \mathbf{R}_{dd} . The adaptive linear filters are continuously adjusted via a set of feedback functions in order to minimize the MSE. Gradient Descent (GD) is used to perform the minimization. The gradient relies on the output autocorrelation \mathbf{R}_{yy} , the cross-correlation between the output and input, \mathbf{R}_{yx_m} , and the desired output autocorrelation \mathbf{R}_{dd} . The autocorrelation sequence $R_{x_mx_m}(\tau)$ of the multi-channel input sequence $x_m(n)$ is given by:

$$R_{x_mx_m}(\tau) = \sum_{n=0}^{N-1} x_m(n)x_m(n-\tau). \quad (17)$$

CS is defined as a multi-input single-output linear filter:

$$y(n) = \sum_{m=0}^{M-1} \mathbf{g}_m^T(n) \mathbf{x}_m(n). \quad (18)$$

The autocorrelation sequence $R_{yy}(\tau)$ of the output signal $y(n)$ is expressed as follows:

$$R_{yy}(\tau) = \sum_{n=0}^{N-1} y(n)y(n-\tau). \quad (19)$$

where N is the number of samples over which autocorrelation is computed, τ is the correlation lag. The scope of CS is to minimize the weighted MSE given by:

$$e(\tau) = W(\tau) (R_{yy}(\tau) - R_{dd}(\tau))^2, \quad (20)$$

where $W(\tau)$ is a real value weight. The larger $W(\tau)$ is, the more relevant the error at a specific lag is. For dereverberation purposes, the LP residual is fed into the correlation shaping processor, and the target output correlation is set to be $R_{dd}(\tau) = \delta(\tau)$. By further exploiting the autocorrelation symmetry, the gradient can be simplified as:

$$\nabla_m(l) = \sum_{\tau>0} W(\tau) R_{yy}(\tau) (R_{yx_m}(l-\tau) + R_{yx_m}(l+\tau)). \quad (21)$$

This gradient is used in the filter update equation:

$$g_m(l, n+1) = g_m(l, n) - \mu \nabla'_m(l), \quad (22)$$

where μ is the learning rate parameter and $\nabla'_m(l)$ is given by:

$$\nabla'_m(l) = \frac{\nabla_m(l)}{\sqrt{\sum_m \sum_l \nabla_m^2(l)}} \quad (23)$$

The dereverberated speech signal is obtained by applying the equalizer $g(l, n)$ to the input signal. Considering that the reverberation time affects significantly audio quality and ASR accuracy, a ‘don’t care’ region is introduced and applied to autocorrelation lags closed to the zeroth lag in order to improve the suppression of long-term components. This region modifies the gradient in Equation 21 and controls the value of the first autocorrelation lag. Details about the CS dereverberation method can be found in our previous publication [3].

4. SYSTEM CONFIGURATION

ASR Baseline: Our system makes use of the challenge ASR baseline [1], which is a state-of-the-art GMM/DNN system based on Kaldi toolkit [40]. The *GMM sub-system* uses 13-dimensional MFCC feature vectors. LDA is used to project a concatenation of 7 consecutive vectors in a sliding window (91 components) to 40 components. The system uses HMMs with 2500 tied triphone states modeled by 15,000 Gaussians. Maximum Likelihood Linear Transformation (MLLT), and feature space Maximum Likelihood Linear Regression (fMLLR) with Speaker Adaptive Training (SAT) are applied. The *DNN sub-system* uses 40-dimensional log Mel-filterbank features instead of MFCCs [4, 29]. It has 7 layers with 2048 neurons each. The input layer has 440 units (5 frames of left and right context). The DNN is trained using the standard procedure: pre-training using Restricted Boltzmann Machines (RBMs), cross entropy training, and sequence discriminative training using the state-level Minimum Bayes Risk (sMBR) criterion. In addition, a baseline signal enhancement is provided, which transforms the multi-channel noisy signal into a single-channel enhanced signal suitable for ASR [1].

Single-channel enhancement: We use BLSTMs for prediction of either speech or noise features with 3 hidden layers. BLSTMs have 128 units per direction. Feed-forward layers with 64 units are inserted after each BLSTM layer in order to reduce the number of parameters. Networks are trained exactly as described in [17] using the BPTT training algorithm. To prevent over-fitting, Gaussian noise with zero mean and 0.1 standard deviation is added to the inputs. Input and target features are standardized to zero mean and unit variance, and delta regression coefficients of the feature contours are added. The sum-squared errors at the output layer per sequence is used as a cost function.

Dereverberation: PEF and CS are performed on top of the SSE. PEF is evaluated using a frame size of 1024 samples. Smaller frame sizes result in less reliable phase estimates causing artifacts and distortions in the reconstructed signal. A frame shift of 10ms is applied. γ was set to 0.01 in order to avoid an aggressive masking that is suitable only in low SNR conditions. m was set to M in order to obtain the geometric mean of the signal and avoid severe speech distortions. CS

is performed by estimating autocorrelation functions on the whole speech segment. We applied 62.5ms long equalizers, a 18.7ms long ‘don’t care’ region and exponential weighting. CS is performed up to $\tau_{max} = 62.5$ ms.

LSTM LM: The standard LM for CHiME-3 is a trigram backoff LM with 5k vocabulary trained on the official LM training data, which consists of around 1.6M sentences (40.5M running words). Training a LSTM LM on this complete data requires impractically huge resources. Therefore, a fraction of the training sentences is selected based on a sentence-level relevance measure with respect to the development set. Thus, a 5-gram backoff LM, called *dev-LM*, is estimated from the development text, then all the sentences of the LM training data are ranked based on their perplexity with this *dev-LM*. A top most 80k sentences are selected with around 2M running words, which is only 5% of the total training data. This procedure allows for estimating the most relevant part of the training text based on domain similarity with the development set. Our LSTM network uses one hidden layer of 300 units. Both input and output layers have 5k units (similar to the vocabulary size). No projection layer is used after the input layer. The long-span probabilities of the LSTM LM are linearly interpolated with a background 5-gram backoff LM trained on the complete training data. The interpolation weight is optimized on the development set. The backoff LMs are estimated with Modified Kneser-Ney (MKN) smoothing using the SRILM toolkit [41].

LSTM Toolkit: In all our experiments, LSTMs are trained and evaluated using our own open-source implementation named CURRENNT (CUDA Recurrent Neural Network Toolkit)¹ [42]. CURRENNT uses Graphical Processing Units (GPUs) to speed up computation. Since in the case of RNNs, parallelization cannot be performed across timesteps due to the temporal dependencies, it parallelizes computations across sequences, for each timestep. This leads to a ‘semi-online’ GD algorithm, where the weights of the network are updated after each batch of parallel sequences.

5. RESULTS

Table 1 shows the WER performance of the GMM and DNN systems for different test sets and training conditions. For a GMM system trained on clean data (original WSJ0 recordings), recognition of noisy test speech shows the poorest performance. However, in this case, the simulated data can be used to approximately predict the performance of the real data. For a baseline enhanced test speech, the performance improved significantly. Yet, a big difference appears between the performance of the real and simulated data; this is most probably due to the limitations of both the acoustic simulation baseline and the enhancement baseline. For SSE of channel 5, performance improvement is only seen on the real eval

¹<http://sourceforge.net/p/currennt>

test set. However, for other test sets, noticeable performance degradations are recorded. This is due to the potential mismatch between the originally clean training conditions and the enhanced test data, as well as the limitation of single channel processing. Applying additional PEF and CS dereverberation to the SSE enhanced test sets leads to significant performance improvements for real data with almost comparable values.

For a GMM system trained on noisy data (non-enhanced channel 5), the performances are comparable after applying each enhancement method to the test sets. In addition, performance similarity is almost kept for real and simulated data.

For a DNN system trained on baseline enhanced data, the best performance is recorded for the baseline enhanced test sets. This is essentially expected due to the direct match between the training and testing conditions. Nevertheless, it is worth observing that, among the other enhancement methods, the CS approach achieves the best performance. Therefore, we train a full DNN system on the CS dereverberated training data. This achieves the best results so far on both real dev and eval test sets. In addition, the simulated data shows a good capability to predict the performance of the real data.

Using a DNN system with an additional N-best rescoring via a LSTM LM interpolated with a background 5-gram back-off LM leads to further improvements in WERs. The N-best rescoring adds around 5% relative WER improvement for the case of baseline enhanced train and test data; and around 4% WER improvement for the case of CS dereverberated train and test data compared to the non-rescored experiments. The best overall results are achieved using a DNN system with additional N-best rescoring via LSTM LM. Around 25% relative WER improvement is achieved compared to the official challenge baseline results (given by line 11 in Table 1). Table 2 shows the detailed WERs for our best system for every environment and every dev/eval test set.

6. CONCLUSIONS

This paper has presented our approach to the 3rd CHiME Separation and Recognition Challenge (CHiME-3), employing single-channel speech enhancement via a fully data-based paradigm using deep BLSTM RNNs. In addition, two multi-channel dereverberation methods called phase-error based filtering and correlation shaping have been applied using the 6-channel audio recordings of the challenge to estimate and filter the reverberation. Furthermore, an improved LM based on LSTM RNNs has been utilized to perform N-best rescoring of the recognition hypotheses. Experiments have been performed according to the official challenge guidelines.

The correlation shaping dereverberation has led to slightly better performance compared to the phase-error based filtering approach. This agrees with the common wisdom that reducing the length of the equalized speaker-to-receiver impulse response can improve the audible quality and ASR accuracy. Both methods have achieved better performance in

Table 1. WER performance of the GMM and DNN systems for real and simulated, dev and eval test sets. Models are trained/tested on: clean, noisy or enhanced noisy data. **enhan**: baseline enhancement (all channels); **SSE5**: single-channel speech enhancement (channel 5); **PEF**: dereverberation via phase-error based filtering (all channels); **CS**: dereverberation via correlation shaping (all channels). **DNN+**: DNN acoustic model + N-best rescoring via LSTM LM.

model	train	test	dev test		eval test	
			real	sim	real	sim
GMM	clean	noisy	54.45	50.56	79.01	63.83
		enhan	41.69	21.88	75.86	25.86
		SSE5	41.83	39.12	66.84	46.96
		PEF	33.91	31.94	57.67	40.92
		CS	35.27	37.84	56.27	46.02
	noisy	noisy	18.71	18.82	33.95	21.92
		enhan	18.59	10.06	32.11	11.30
		SSE5	21.12	20.16	36.10	21.50
		PEF	16.51	16.26	28.79	19.55
		CS	17.03	18.81	27.74	20.92
DNN	enhan	enhan	17.76	8.67	32.71	10.79
		SSE5	33.67	29.63	55.96	35.75
		PEF	25.95	30.47	40.76	42.98
		CS	23.49	29.87	36.51	40.77
	CS	CS	14.08	15.34	25.53	22.32
DNN+	enhan	enhan	16.71	7.82	30.81	10.11
	CS	CS	13.71	14.56	24.38	21.45

Table 2. Detailed WER performance for the best system with the least WER on real dev test set (row 15 of Table 1).

environment	dev test		eval test	
	real	sim	real	sim
BUS	15.33	11.71	30.68	17.20
CAF	14.48	19.51	27.31	24.24
PED	9.96	12.08	21.26	22.38
STR	15.07	14.94	18.29	21.98

comparison to the single-channel enhancement alone. Moreover, the application of N-best rescoring via the LSTM LM has led to improved performance.

Further improvements are possible by improving the back-end recognition using, for example, BLSTM based acoustic models, and discriminatively trained GMMs. In addition, the LSTM LM needs to be trained on more data from the official dataset. It is also worth trial to train two full DNN systems using the single-channel enhanced and dereverberated data via phase-error based filtering in order to draw a fair comparison among the employed enhancement techniques.

7. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Unions Horizon 2020 Programme through the Research Innovation Action No. 645378 (ARIA-VALUSPA); and from the German Federal Ministry of Education, Science, Research and Technology (BMBF) under grant agreement No. 16SV7213 (EmotAsS).

8. REFERENCES

- [1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Submitted to IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, Arizona, USA, Dec 2015.
- [2] T. Virtanen, R. Singh, and B. Raj, *Techniques for Noise Robustness in automatic speech recognition*, John Wiley & Sons, Ltd, Chichester, UK, Oct. 2012.
- [3] J. Geiger, E. Marchi, B. Schuller, and G. Rigoll, "The TUM system for the REVERB Challenge: Recognition of Reverberated Speech using Multi-Channel Correlation Shaping Dereverberation and BLSTM Recurrent Neural Networks," in *Proc. REVERB Workshop, held in conjunction with ICASSP 2014 and HSCMA 2014*, Florence, Italy, May 2014, pp. 1–8.
- [4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.
- [5] E. Arisoy, T. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in *NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montreal, Canada, June 2012, pp. 20 – 28.
- [6] A. El-Desoky, H.-K. J. Kuo, L. Mangu, and H. Soltau, "Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, BC, Canada, May 2013.
- [7] T. Mikolov, *Statistical Language Models based on Neural Networks*, Ph.D. thesis, Brno University of Technology, 2012.
- [8] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, Portland, OR, USA, Sept. 2012.
- [9] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsalo, "A class of neural networks for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 8, no. 3, pp. 486–504, May 1997.
- [10] Y. Tan, J. Wang, and J.M. Zurada, "Nonlinear blind source separation using a radial basis function network," *Neural Networks, IEEE Transactions on*, vol. 12, no. 1, pp. 124–134, Jan 2001.

- [11] A. L. Maas, T. M. O’Neil, A. Y. Hannun, and A. Y. Ng, “Recurrent Neural Network Feature Enhancement: The 2nd CHiME Challenge,” in *Proc. The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013*, Vancouver, Canada, June 2013, IEEE, pp. 79–80, IEEE.
- [12] M. Wöllmer, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, “Feature Enhancement by Bidirectional LSTM Networks for Conversational Speech Recognition in Highly Non-Stationary Noise,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, IEEE, pp. 6822–6826, IEEE.
- [13] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, “The Munich Feature Enhancement Approach to the 2013 CHiME Challenge Using BLSTM Recurrent Neural Networks,” in *Proc. The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013*, Vancouver, Canada, June 2013, IEEE, pp. 86–90, IEEE.
- [14] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder,” in *Interspeech*, Lyon, France, August 2013, pp. 436–440.
- [15] B. Xia and C. Bao, “Speech enhancement with weighted denoising auto-encoder,” in *Interspeech*, Lyon, France, August 2013, pp. 3444–3448.
- [16] A. Narayanan and D. Wang, “Ideal ratio mask estimation using deep neural networks for robust speech recognition,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 7092–7096.
- [17] F. Weninger, F. Eyben, and B. Schuller, “Single-Channel Speech Separation With Memory-Enhanced Recurrent Neural Networks,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 3737–3741.
- [18] J.B. Allen and D.A. Berkley, “Image method for efficiently simulating smallroom acoustics,” *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.
- [19] P.M. Peterson, “Simulating the response of multiple microphones to a single acoustic source in a reverberant room,” *The Journal of the Acoustical Society of America*, vol. 80, no. 5, pp. 1527–1529, 1986.
- [20] B. Yegnanarayana and P.S. Murthy, “Enhancement of reverberant speech using lp residual signal,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 267–281, May 2000.
- [21] S. Griebel and M. Brandstein, “Wavelet transform extrema clustering for multi-channel speech dereverberation,” *Proc. IEEE IWAENC*, pp. 27–30, 1999.
- [22] D. Ward and M. Brandstein, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, Berlin, Germany, 2001.
- [23] G. Xu, H. Liu, L. Tong, and T. Kailath, “A least-squares approach to blind channel identification,” *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2982–2993, Dec 1995.
- [24] S. Gannot and M. Moonen, “Subspace methods for multimicrophone speech dereverberation,” *EURASIP J. Appl. Signal Process.*, vol. 2003, pp. 1074–1090, Jan. 2003.
- [25] Y.A. Huang, J. Benesty, and J. Chen, “A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 882–895, Sept 2005.
- [26] C. Y-K. Lai and P. Aarabi, “Multiple-microphone time-varying filters for robust speech recognition,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, vol. 1, pp. 230–233.
- [27] P. Aarabi and G. Shi, “Phase-based dual-microphone robust speech enhancement,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1763–1773, Aug 2004.
- [28] B. W. Gillespie and L.E. Atlas, “Strategies for improving audible quality and speech recognition accuracy of reverberant speech,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, April 2003, vol. 1, pp. 676–679.
- [29] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.
- [30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, no. 323, pp. 533–536, 1986.
- [31] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.
- [32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [33] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451 – 2471, 1999.
- [34] F. A. Gers, *Long short-term memory in recurrent neural networks*, Ph.D. thesis, Department of Computer Science, Swiss Federal Institute of Technology, Lausanne, EPFL, Switzerland, 2001.
- [35] A. Mohamed, G. Dahl, and G. E. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, Whistler, BC, Canada, 2009.
- [36] T. Sainath, B. Kingsbury, and B. Ramabhadran, "Improvements in using deep belief networks for large vocabulary continuous speech recognition," Technical report, IBM, Speech and Language Algorithms Group, 2012.
- [37] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2013, pp. 273 – 278.
- [38] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, Jul 2001.
- [39] Jürgen T. Geiger, Felix Weninger, Antti Hurmalainen, Jort F. Gemmeke, Martin Wöllmer, Björn Schuller, Gerhard Rigoll, and Tuomas Virtanen, "The TUM+TUT+KUL approach to the CHiME challenge 2013: Multi-stream ASR exploiting BLSTM networks and sparse NMF," in *Proc. The 2nd CHiME Workshop on Machine Listening in Multisource Environments held in conjunction with ICASSP 2013*, Vancouver, Canada, June 2013, IEEE, pp. 25–30, IEEE.
- [40] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011, IEEE Signal Processing Society.
- [41] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, Colorado, USA, Sept. 2002, vol. 2, pp. 901 – 904.
- [42] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENNT – the Munich open-source CUDA Recurrent Neural Network Toolkit," *Journal of Machine Learning Research*, vol. 15, no. 99, Oct. 2014.